Classificação Externa: Geração de Partições Classificadas

Vanessa Braganholo

Baseado no Material de: Inhaúma Neves Ferraz (IC/UFF)

Importância da Ordenação/Classificação

- Vimos até agora várias operações envolvendo arquivos sequenciais
  - Todas elas exigem que os arquivos estejam ordenados para melhor desempenho (ex.: consulta, atualização)

Ordenação em Memória Principal

- Vimos também que uma possível forma de ordenar um arquivo é:
  1. ler todos os registros
  2. armazená-los em memória principal
  3. ordenar os registros
  4. gravá-los em um arquivo de saída

Mas...

- E quando os registros não cabem na memória?

Tipos de Classificação

- **Classificação interna**: utilização exclusiva de memória principal
  - Todos os registros do arquivo cabem em memória principal
- **Classificação externa**: utilização de memória secundária
  - Há mais registros a serem classificados do que é possível manter na memória principal em qualquer momento

ATENÇÃO: Nessa disciplina usamos o termo **classificação** como sinônimo de **ordenação**

Conceito de Classificação Externa

- Na classificação externa o parâmetro fundamental é o número de operações de entrada e saída
  - Deve ser o menor possível

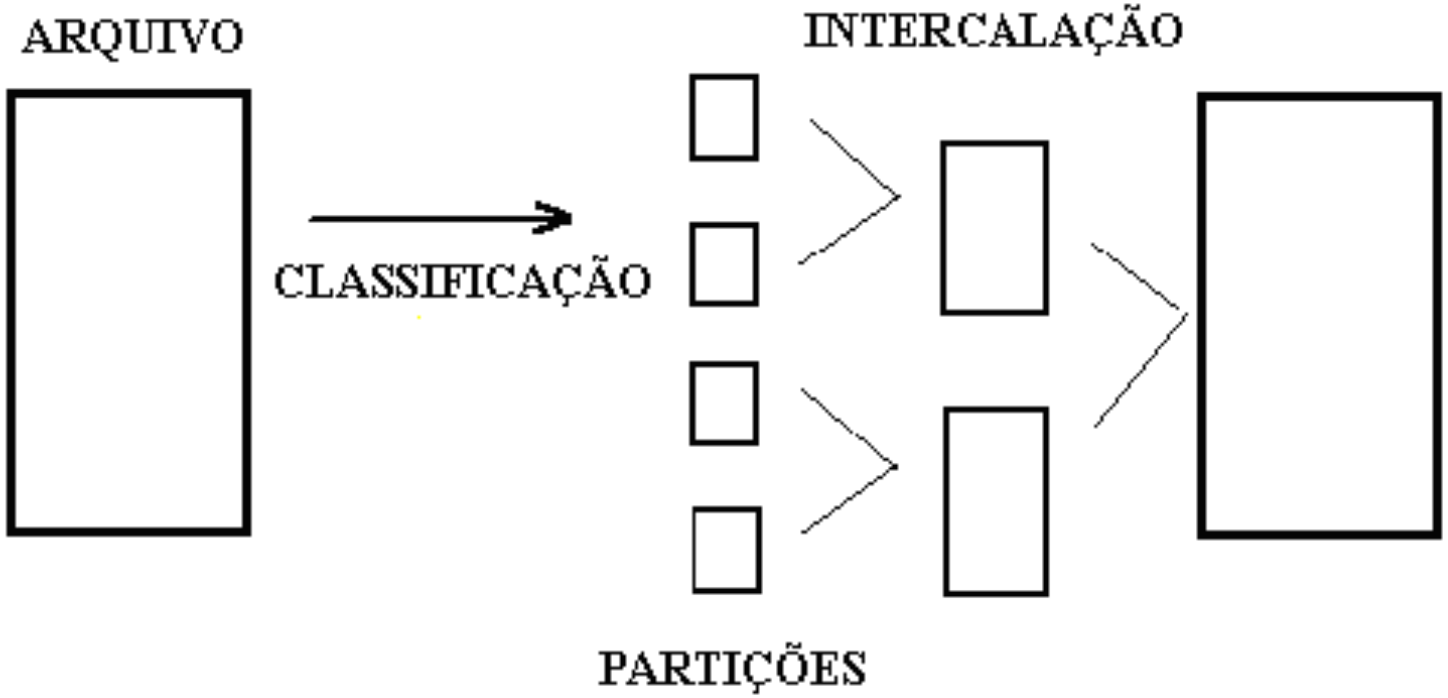
Discussão

- Como poderíamos resolver o problema de ordenar um arquivo muito grande, que não cabe em memória?

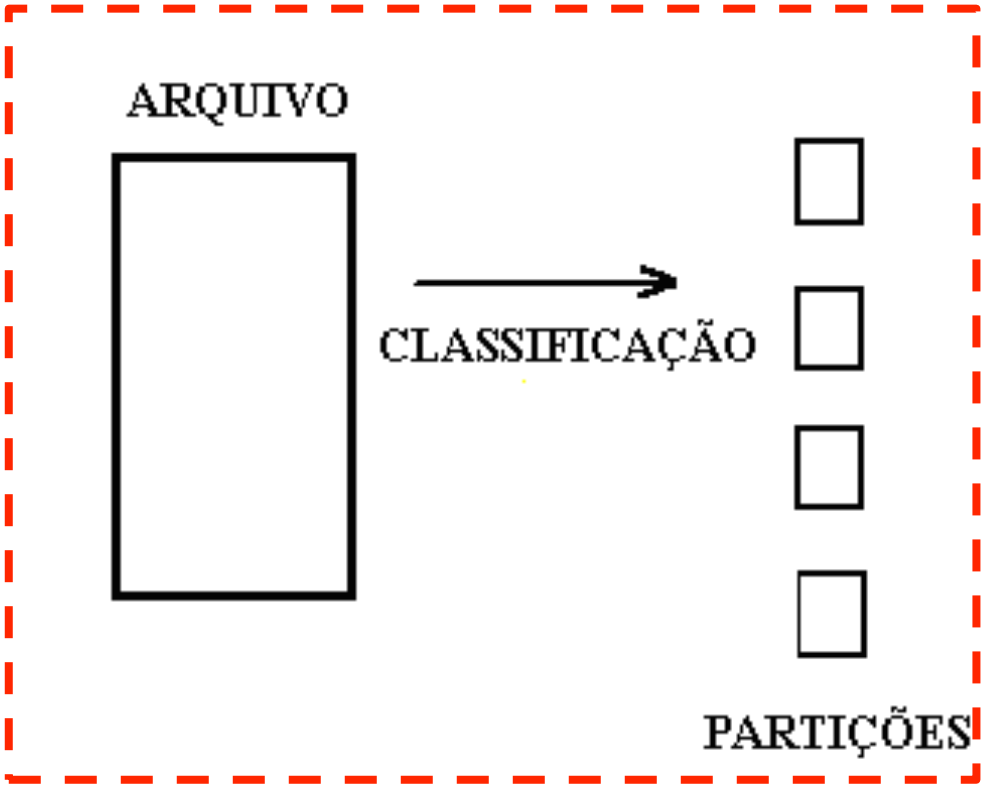
Ideia básica da Classificação Externa

- A Classificação Externa divide os arquivos em pequenas frações que são ordenadas e intercaladas em duas etapas:
  - Classificação
  - Intercalação

Modelo da Classificação Externa



Na aula de hoje: Etapa de Classificação



Etapa de Classificação

- **Partição**: sequencia ordenada de *n* registros.
- Registros são lidos de arquivos de entrada (não ordenados) e/ou de partições (ordenadas)
- Estes registros são ordenados e gravados em arquivos de saída ou partições ordenadas

Geração de Partições Classificadas

Métodos do Estágio de Classificação

- Métodos
  - Classificação interna
  - Seleção com substituição
  - Seleção natural
- Considera-se que a memória principal tenha capacidade para armazenar *M* registros do arquivo a classificar

Classificação Interna

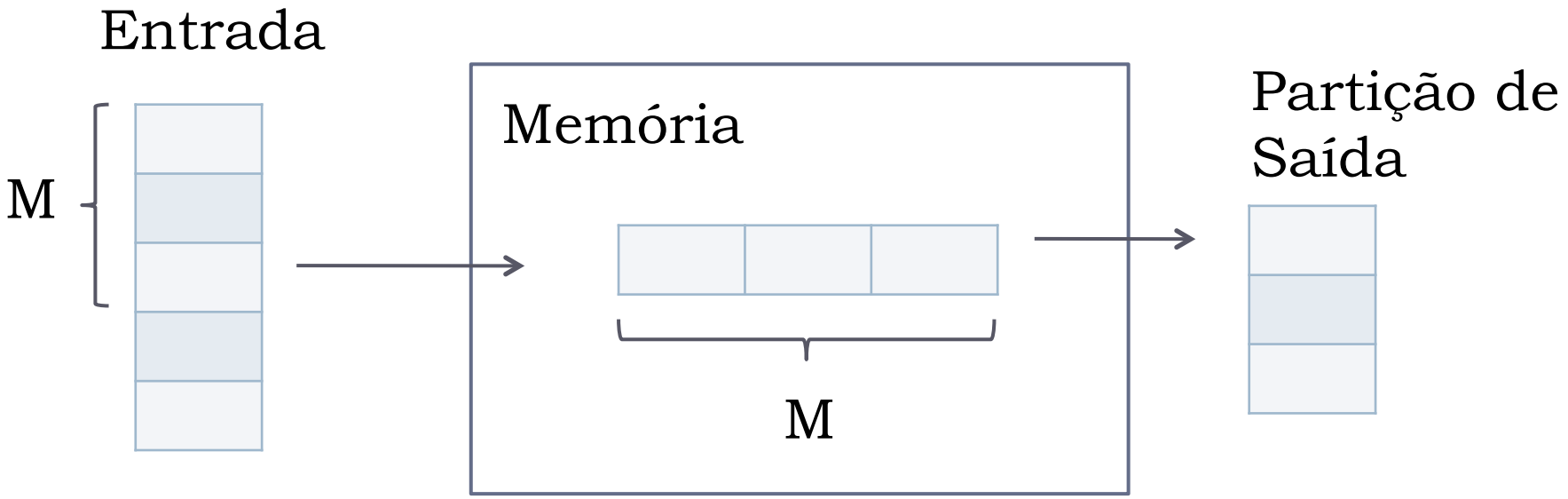
Classificação Interna

- Critério fundamental de eficiência da classificação interna: número de comparações entre chaves de registros
- Consiste na leitura de *M* registros para a memória, classificação desses registros por qualquer processo de classificação interna e gravação desses registros classificados em uma partição
- Todas as partições classificadas contêm *M* registros, exceto, talvez, a última

Processos de classificação interna

- Troca: bubble sort, shaker sort, quick sort
- Seleção: direta, heap sort,
- Inserção: simples, shell sort
- Outros: merge sort, etc.

Visão geral da Geração de partições Classificadas



Exemplo

- Chaves do arquivo a ordenar
  - (Sequência de leitura: 29, 14, 76,…)
- Assumir que na memória cabem 6 registros simultaneamente

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

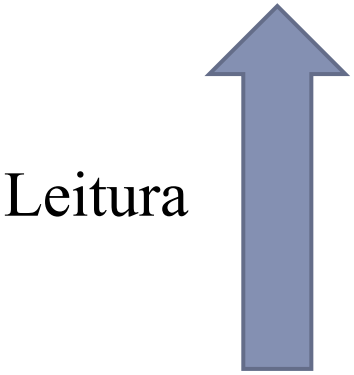
Exemplo

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

Exemplo

Memória Principal

29	14	76	75	59	6



29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

Exemplo

Memória Principal

29	14	76	75	59	6

Ordenação

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

Exemplo

Memória Principal

6	14	29	69	75	76

Ordenação

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

Exemplo

Memória Principal

6	14	29	69	75	76

Partição 1 (em disco) ordenada

6	14	29	69	75	76

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

Exemplo

Partição 1 (em disco) ordenada

6	14	29	69	75	76

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

	Área de trabalho							Partições obtidas					
Memória	29	14	76	75	59	6		6	14	29	59	75	76
Memória	7	74	48	46	10	18		7	10	18	46	48	74
Memória	56	20	26	4	21	65		4	20	21	26	56	65
Memória	22	49	11	16	8	15		8	11	15	16	22	49
Memória	5	19	50	55	25	66		5	19	25	50	55	66
Memória	57	77	12	30	17	9		9	12	17	30	57	77
Memória	54	78	43	38	51	32		32	38	43	51	54	78
Memória	58	13	73	79	27	1		1	13	27	58	73	79
Memória	3	60	36	47	31			3	31	36	47	60	

Seleção com Substituição

Seleção com substituição

- Aproveita a possível classificação parcial do arquivo de entrada

Seleção com substituição: Algoritmo

1. Ler M registros do arquivo para a memória
2. Selecionar, no array em memória, o registro r com menor chave
3. Gravar o registro r na partição de saída
4. Substituir, no array em memória, o registro r pelo próximo registro do arquivo de entrada
5. Caso a chave deste último seja menor do que a chave recém gravada, considerá-lo congelado e ignorá-lo no restante do processamento
6. Caso existam em memória registros não congelados, voltar ao passo 2
7. Caso contrário:
   - fechar a partição de saída
   - descongelar os registros congelados
   - abrir nova partição de saída
   - voltar ao passo 2

Exemplo

- Chaves do arquivo a ordenar
  - (Sequência de leitura: 29, 14, 76,…)
- Assumir que na memória cabem 6 registros simultaneamente

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31	80						

	Área de trabalho							Partições obtidas													
Registros	1	2	3	4	5	6		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3ª substituição						20																
2ª substituição	10	18				74																
1ª substituição	46	48	4	26	56	7																
Memória	29	14	76	75	59	6		6	7	14	29	46	48	59	74	75	76					
								A 1ª partição ficou com 10 registros													
2ª substituição	19	16	11			15																
1ª substituição	65	22	21	8	5	49																
Memória	10	18	4	26	56	20		4	10	18	20	21	22	26	49	56	65					
								A 2ª partição ficou com 10 registros													
3ª substituição	43																					
2ª substituição	78	9	12	17	30	54																
1ª substituição	77	57	25	55	50	66																
Memória	19	16	11	8	5	15		5	8	11	15	16	19	25	50	55	57	66	77	78		
								A 3ª partição ficou com 13 registros													
3ª substituição		60																				
2ª substituição	36	73	27	13	3																	
1ª substituição	79	38	51	32	58	1																
Memória	43	9	12	17	30	54		9	12	17	30	32	38	43	51	54	58	73	79			
								A 4ª partição ficou com 12 registros													
1ª substituição				80	31	47																
Memória	36	60	27	13	3	1		1	3	13	27	31	36	47	60	80						
								A 5ª partição ficou com 9 registros													

Legenda

Registros congelados		Divisão de regiões na tabela	

Tamanho das partições geradas

- Em média, o tamanho das partições obtidas pelo processo de seleção com substituição é de 2 * M

Seleção Natural

Seleção Natural

- Desvantagem da seleção com substituição: no final da partição grande parte do espaço em memória principal está ocupado com registros congelados
- Na seleção natural, reserva-se um espaço de **memória secundária** (o reservatório) para abrigar os registros congelados num processo de substituição
- A formação de uma partição se encerra quando o reservatório estiver cheio ou quando terminarem os registros de entrada
- Para a memória comportando *M* registros supõe-se um reservatório comportando *n* registros
- Para *M* = *n* o comprimento médio das partições é de M * e, onde e = 2,718... .

Seleção Natural: Algoritmo

1. Ler M registros do arquivo para a memória
2. Selecionar, no array em memória, o registro r com menor chave
3. Gravar o registro r na partição de saída
4. Substituir, no array em memória, o registro r pelo próximo registro do arquivo de entrada
5. Caso a chave deste último seja menor do que a chave recém gravada, gravá-lo no reservatório e substituir, no array em memória, o registro r pelo próximo registro do arquivo de entrada
6. Caso ainda exista espaço livre no reservatório, voltar ao passo 2
7. Caso contrário:
   - fechar a partição de saída
   - copiar os registros do reservatório para o array em memória
   - esvaziar o reservatório
   - abrir nova partição de saída
   - voltar ao passo 2

Exemplo

- Chaves do arquivo a ordenar
  - (Sequência de leitura: 29, 14, 76,…)
- Assumir que na memória cabem 6 registros simultaneamente (M = 6), e que o tamanho do reservatório também é 6 (n = 6)

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31	80						

	Área de trabalho						Partições obtidas														
Registros	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2ª substituição	56					74															
1ª substituição	46	48				7															
Memória	29	14	76	75	59	6	6	7	14	29	46	48	56	59	74	75	76				
Reservatório	10	18	20	26	4	21															
							A 1ª partição ficou com 11 registros														
1ª substituição	22	49			65																
Memória	10	18	20	26	4	21	4	10	18	20	21	22	26	49	65						
Reservatório	11	16	8	15	5	19															
							A 2ª partição ficou com 9 registros														
3ª substituição	54																				
2ª substituição	30				78																
1ª substituição	25	57	55	66	50	77															
Memória	11	16	8	15	5	19	5	8	11	15	16	19	25	30	50	54	55	57	66	77	78
Reservatório	12	17	9	43	38	51															
							A 3ª partição ficou com 15 registros														
2ª substituição			79																		
1ª substituição	58	73	32	47	60																
Memória	12	17	9	43	38	51	9	12	17	32	38	43	47	51	58	60	73	79			
Reservatório	13	27	1	3	36	31															
							A 4ª partição ficou com 12 registros														
1ª substituição			80																		
Memória	13	27	1	3	36	31	1	3	13	27	31	36	80								
Reservatório							A 5ª partição ficou com 7 registros														

Comparação dos Processos

- A classificação interna gera as menores partições, o que implica em mais arquivos a intercalar
- Os processos de seleção geram partições maiores, reduzindo o tempo total de processamento
- A seleção natural sofre o ônus adicional de utilizar mais operações de entrada e saída (devido ao reservatório estar em memória secundária)

Exercício 1

- Gerar partições classificadas segundo o método de Seleção com Substituição para a seguinte situação
- Assumir que na memória cabem 7 registros simultaneamente
- Arquivo a ordenar

30	14	15	75	32	6	5	81	48	41	87	18
56	20	26	4	21	65	22	49	11	16	8	12

Exercício 2

- Repetir o exercício anterior, agora utilizando o método de Seleção Natural
- Assumir que na memória cabem 7 registros simultaneamente. Tamanho do reservatório = 7.
- Arquivo a ordenar

30	14	15	75	32	6	5	81	48	41	87	18
56	20	26	4	21	65	22	49	11	16	8	12

Exercício 3

- Implementar um dos 2 métodos de geração de partições vistos na aula de hoje (Seleção com Substituição ou Seleção Natural)